

What is claimed is:

1. In an iterative decoder, an iterative decoding method for decoding received digital data elements representing source data elements coded according to a turbo coding

5 scheme, the iterative decoding method comprising:

computing a set of branch metrics for the received digital data elements based upon at least one received digital data element;

computing a set of forward recursive metrics based upon the set of branch metrics according to an approximation:

$$10 \quad A_k(m) = \log[\alpha_k(m)] = \max_{m'} \{ \Gamma(u_k, c_k, m', m) + A_{k-1}(m') \} - H_{A_k};$$

computing a set of backward recursive metrics based upon the set of branch metrics according to an approximation:

$$B_k(m') = \log[\beta_k(m')] = \max_m \{ \Gamma(u_k, c_k, m', m) + B_{k+1}(m) \} - H_{B_k}; \text{ and}$$

15 computing a set of output extrinsic Log Likelihood Ratio (LLR) values based upon the set of backward metrics and the set of forward metrics according to an equation:

$$\Lambda(d_k) = \log \left[ \sum_{e: u(e)=d_k=+1} e^{\{A_{k-1}(m') + \Gamma(c_k, m', m) + B_k(m)\}} \right] - \log \left[ \sum_{e: u(e)=d_k=0} e^{\{A_{k-1}(m') + \Gamma(c_k, m', m) + B_k(m)\}} \right]$$

2. The iterative decoding method of claim 1, wherein computing the set of output  
20 extrinsic LLR values comprises:

maintaining a table of logarithm values; and

using the table of logarithm values to determine the logarithm of the value  $L(d_k)$  where .

$$25 \quad L(d_k) = \log \left[ \sum_{e: u(e)=d_k} e^{\{A_{k-1}(m') + \Gamma(c_k, m', m) + B_k(m)\}} \right]$$

and  $d_k$  is either +1 or 0.

3. The iterative decoding method of claim 2, wherein using the table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  comprises:

obtaining the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  directly from the table of logarithm values.

5

4. The iterative decoding method of claim 2, wherein using the table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  comprises:

obtaining, from the table of logarithm values, a logarithm of a value less than  $L(d_k = +1)$  ;

10 obtaining, from the table of logarithm values, a logarithm of a value less than  $L(d_k = 0)$  ;

obtaining, from the table of logarithm values, a logarithm of a value greater than  $L(d_k = +1)$  ; obtaining, from the table of logarithm values, a logarithm of a value greater than  $L(d_k = 0)$  ; and

15 deriving the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and deriving the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$  .

20 5. The iterative decoding method of claim 4, wherein deriving the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and deriving the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$  comprises:

25 extrapolating the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and

extrapolating the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$ .

6. The iterative decoding method of claim 1, wherein computing the set of backward recursive metrics comprises:

using a sliding window for processing less than the entirety of received digital data elements.

7. The iterative decoding method of claim 6, wherein using a sliding window for processing less than the entirety of received digital data elements comprises:

initializing the set of backward recursive metrics with equal probabilities.

8. The iterative decoding method of claim 6, wherein using a sliding window for processing less than the entirety of received digital data elements comprises:

initializing the set of backward recursive metrics with the set of forward recursive metrics.

9. An iterative decoder for decoding received digital data elements representing source data elements coded according to a turbo coding scheme, the decoder comprising:

branch metric logic operably coupled to compute a set of branch metrics for the received digital data elements based upon at least one received digital data element;

forward recursive metric logic operably coupled to compute a set of forward recursive metrics based upon the set of branch metrics according to an approximation:

$$A_k(m) = \log[\alpha_k(m)] = \max_{m'} \{ \Gamma(u_k, c_k, m', m) + A_{k-1}(m') \} - H_{A_k};$$

backward recursive metric logic operably coupled to compute a set of backward recursive metrics based upon the set of branch metrics according to an approximation:

$$B_k(m') = \log[\beta_k(m')] = \max_m \{ \Gamma(u_k, c_k, m', m) + B_{k+1}(m) \} - H_{B_k}; \text{ and}$$

extrinsic logic operably coupled to compute a set of output extrinsic Log Likelihood Ratio (LLR) values based upon the set of backward metrics and the set of forward metrics according to an equation:

$$\Lambda(d_k) = L(d_k = +1) - L(d_k = 0).$$

10. The iterative decoder of claim 9, wherein the extrinsic logic comprises:  
logic for using a table of logarithm values to determine the logarithm of the values  
5  $L(d_k = +1)$  and  $L(d_k = 0)$ .
11. The iterative decoder of claim 10, wherein the logic for using the table of  
logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$   
comprises:  
10 logic for obtaining the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  from the  
table of logarithm values.
12. The iterative decoder of claim 10, wherein the logic for using the table of  
logarithm values to determine the logarithm of the value  $L(d_k = +1)$  and  $L(d_k = 0)$   
15 comprises:  
logic for obtaining, from the table of logarithm values, a logarithm of a value less  
than  $L(d_k = +1)$ ;  
logic for obtaining, from the table of logarithm values, a logarithm of a value  
greater than  $L(d_k = +1)$ ;  
20 logic for obtaining, from the table of logarithm values, a logarithm of a value less  
than  $L(d_k = 0)$ ;  
logic for obtaining, from the table of logarithm values, a logarithm of a value  
greater than  $L(d_k = 0)$ ; and  
logic for deriving the logarithm of the value  $L(d_k = +1)$  from the logarithm of the  
25 value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and for  
deriving the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  
 $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$ .

13. The iterative decoder of claim 12, wherein the logic for deriving the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  from the logarithm of the values less than  $L(d_k = +1)$  and  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = +1)$  and  $L(d_k = 0)$  comprises:
- 5        logic for extrapolating the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and logic for extrapolating the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$ .
- 10    14. The iterative decoder of claim 9, wherein the backward recursive metric logic comprises:
- logic for using a sliding window for processing less than the entirety of received digital data elements.
- 15    15. The iterative decoder of claim 14, wherein the logic for using a sliding window for processing less than the entirety of received digital data elements comprises:
- logic for initializing the set of backward recursive metrics with equal probabilities.
- 20    16. The iterative decoder of claim 14, wherein the logic for using a sliding window for processing less than the entirety of received digital data elements comprises:
- logic for initializing the set of backward recursive metrics with the set of forward recursive metrics.
- 25    17. An apparatus for decoding received digital data elements representing source data elements coded according to a turbo coding scheme, the apparatus comprising:
- a first iterative decoder;
- a second iterative decoder;

an interleaver operably coupled to receive first output extrinsic Log Likelihood Ratio (LLR) values from the first iterative decoder and to provide interleaved output extrinsic Log Likelihood Ratio (LLR) values to the second iterative decoder; and

5 a deinterleaver operably coupled to receive second output extrinsic Log Likelihood Ratio (LLR) values from the second iterative decoder and to provide deinterleaved output extrinsic Log Likelihood Ratio (LLR) values to the first iterative decoder, wherein each iterative decoder comprises:

branch metric logic operably coupled to compute a set of branch metrics for the received digital data elements based upon at least one received digital data element;

10 forward recursive metric logic operably coupled to compute a set of forward recursive metrics based upon the set of branch metrics according to an approximation:

$$A_k(m) = \log[\alpha_k(m)] = \max_{m'} \{ \Gamma(u_k, c_k, m', m) + A_{k-1}(m') \} - H_{A_k};$$

backward recursive metric logic operably coupled to compute a set of backward recursive metrics based upon the set of branch metrics according to an approximation:

15 
$$B_k(m') = \log[\beta_k(m')] = \max_m \{ \Gamma(u_k, c_k, m', m) + B_{k+1}(m) \} - H_{B_k}; \text{ and}$$

extrinsic logic operably coupled to compute a set of output extrinsic Log Likelihood Ratio (LLR) values based upon the set of backward metrics and the set of forward metrics according to an equation:

$$\Lambda(d_k) = L(d_k = +1) - L(d_k = 0).$$

20

18. The apparatus of claim 17, further comprising a table of logarithm values, wherein the extrinsic logic comprises:

logic for using the table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$ .

25

19. The apparatus of claim 18, wherein the logic for using the table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  comprises:

logic for obtaining the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  from the table of logarithm values.

30

20. The apparatus of claim 18, wherein the logic for using the table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  comprises:

logic for obtaining, from the table of logarithm values, a logarithm of a value less than  $L(d_k = +1)$ ;

5 logic for obtaining, from the table of logarithm values, a logarithm of a value greater than  $L(d_k = +1)$ ;

logic for obtaining, from the table of logarithm values, a logarithm of a value less than  $L(d_k = 0)$ ;

10 logic for obtaining, from the table of logarithm values, a logarithm of a value greater than  $L(d_k = 0)$ ; and

logic for deriving the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and logic for deriving the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$ .

15

21. The apparatus of claim 20, wherein the logic for deriving the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$  comprises:

20 logic for extrapolating the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and logic for extrapolating the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$ .

25 22. The apparatus of claim 17, wherein the backward recursive metric logic comprises:

logic for using a sliding window for processing less than the entirety of received digital data elements.

23. The apparatus of claim 22, wherein the logic for using a sliding window for processing less than the entirety of received digital data elements comprises:

5 logic for initializing the set of backward recursive metrics with equal probabilities.

24. The apparatus of claim 22, wherein the logic for using a sliding window for processing less than the entirety of received digital data elements comprises:

10 logic for initializing the set of backward recursive metrics with the set of forward recursive metrics.

25. An apparatus comprising a digital storage medium having embodied therein a program for controlling a programmable logic device to decode received digital data elements representing source data elements coded according to a turbo coding scheme,  
15 the program comprising:

branch metric logic programmed to compute a set of branch metrics for the received digital data elements based upon at least one received digital data element;

forward recursive metric logic programmed to compute a set of forward recursive metrics based upon the set of branch metrics according to an approximation:

20 
$$A_k(m) = \log[\alpha_k(m)] = \max_m \{ \Gamma(u_k, c_k, m', m) + A_{k-1}(m') \} - H_{A_k};$$

backward recursive metric logic programmed to compute a set of backward recursive metrics based upon the set of branch metrics according to an approximation:

$$B_k(m') = \log[\beta_k(m')] = \max_m \{ \Gamma(u_k, c_k, m', m) + B_{k+1}(m) \} - H_{B_k}; \text{ and}$$

extrinsic logic programmed to compute a set of output extrinsic Log Likelihood  
25 Ratio (LLR) values based upon the set of backward metrics and the set of forward metrics according to an equation:

$$\Lambda(d_k) = L(d_k = +1) - L(d_k = 0).$$

26. The apparatus of claim 25, wherein the extrinsic logic comprises:



logic for using a table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$ .

27. The apparatus of claim 26, wherein the logic for using the table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  comprises:

logic for obtaining the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  from the table of logarithm values.

28. The apparatus of claim 26, wherein the logic for using the table of logarithm values to determine the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  comprises:

logic for obtaining, from the table of logarithm values, a logarithm of a value less than  $L(d_k = +1)$ ;

logic for obtaining, from the table of logarithm values, a logarithm of a value greater than  $L(d_k = +1)$ ;

15 logic for obtaining, from the table of logarithm values, a logarithm of a value less than  $L(d_k = 0)$ ;

logic for obtaining, from the table of logarithm values, a logarithm of a value greater than  $L(d_k = 0)$ ; and

20 logic for deriving the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and logic for deriving the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$ .

29. The apparatus of claim 28, wherein the logic for deriving the logarithm of the values  $L(d_k = +1)$  and  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$  comprises:

logic for extrapolating the logarithm of the value  $L(d_k = +1)$  from the logarithm of the value less than  $L(d_k = +1)$  and the logarithm of the value greater than  $L(d_k = +1)$  and logic for extrapolating the logarithm of the value  $L(d_k = 0)$  from the logarithm of the value less than  $L(d_k = 0)$  and the logarithm of the value greater than  $L(d_k = 0)$ .

5

30. The apparatus of claim 25, wherein the backward recursive metric logic comprises:

logic for using a sliding window for processing less than the entirety of received digital data elements.

10

31. The apparatus of claim 30, wherein the logic for using a sliding window for processing less than the entirety of received digital data elements comprises:

logic for initializing the set of backward recursive metrics with equal probabilities.

15

32. The apparatus of claim 30, wherein the logic for using a sliding window for processing less than the entirety of received digital data elements comprises:

logic for initializing the set of backward recursive metrics with the set of forward recursive metrics.

20

33. The apparatus of claim 25, wherein the programmable logic device comprises one of:

a microprocessor;

a digital signal processor; and

25

a field programmable gate array.